Allasso Consulting, LLC

IBM HACMP for AIX 5L
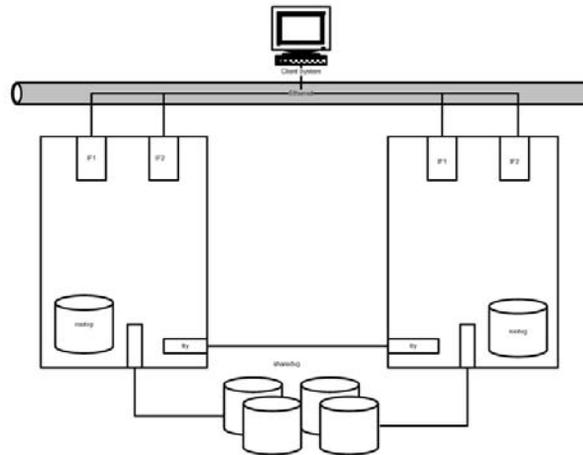
A discussion of design principles for cluster planning.

Author: Chuck Shoecraft

Introduction

HACMP (High Availability Cluster Multi Processing) is IBM's offering for supplying High Availability in servers running AIX or Linux. This document will discuss the design of HACMP for the AIX environment, and the issues that must be addressed in the planning process in order to produce a cluster which will effectively provide highly available computer services. The Linux environment is new in Version 5.4 of HACMP and has a limited set of the HACMP functions available with AIX. Linux and AIX cannot be joined in the same cluster.

HACMP has evolved since its first availability in the late 1980s and is expected to continue to add new function as the System P adds new features relevant to high availability issues. From the beginning HACMP has been a product bound tightly to the AIX operating system, using AIX features including: the Object Data Manager (ODM), a configuration repository; the System Resource Controller (SRC), a subsystem manager; the Logical Volume Manager (LVM), a manager and organizer of data into Volume Groups which can be transported intact between systems; and, most recently, Reliable Scalable Cluster Technology (RSCT), a suite of subsystems providing the facilities to combine systems into clusters for cooperative work, group management and communication capabilities, and resource management function.

HACMP Functional Overview

HACMP is enabling software that makes it possible to provide automatic application continuation in the event of component or system (referred to as node) failure and, in the newer versions, the additional capability to relocate an application based on resource availability. In the original versions, HACMP monitored node, network and network interface health and, in the event of a change in status of any of these, would take appropriate action to reconfigure the cluster. Later versions have expanded the scope of items that can be monitored and responded to.

The basic function of HACMP is accomplished through a Cluster Manager (binary) and a large complement of shell scripts known as Recovery Programs and Event Scripts. Based on information received from RSCT, the Cluster Manager determines that an event (change in cluster status) has occurred and executes the appropriate scripts to reconfigure the cluster so that the application being made highly available will continue to function. The user community will experience an outage of the application for a short period of time and then continue to have access to the service. The time period involved is dependent on many variables and may be as short as the seconds necessary to move an IP address to a different NIC (Network Interface Card) or as long as necessary to validate the status of an entire data structure, mount the file systems and start the application on a different system.

HACMP is an amazingly effective tool for constructing a highly available environment. It is not an automatic provider of high availability. It requires a clear understanding of the application requirements, careful planning and construction of the Cluster Topology (network infrastructure), scripting to provide application start and stop function and assurance of correct system environment, cluster definition, good documentation of cluster configuration and rigorous testing. It accomplishes its high availability goals through the use of the principle of elimination of single points of failure. There is a minimal requirement for what items must be considered in this principle, but it can be taken as far as the loss prevention justification can warrant. This idea will be further explored in a later section.

HACMP configuration is organized in three basic parts: Topology, Resources and Resource Groups. Following sections of this document will discuss each of the basic parts of a cluster design.

Cluster Topology

Cluster topology is the view of the HACMP Cluster from a networking and connection perspective. The diagram below will serve as a guide to the following discussion.
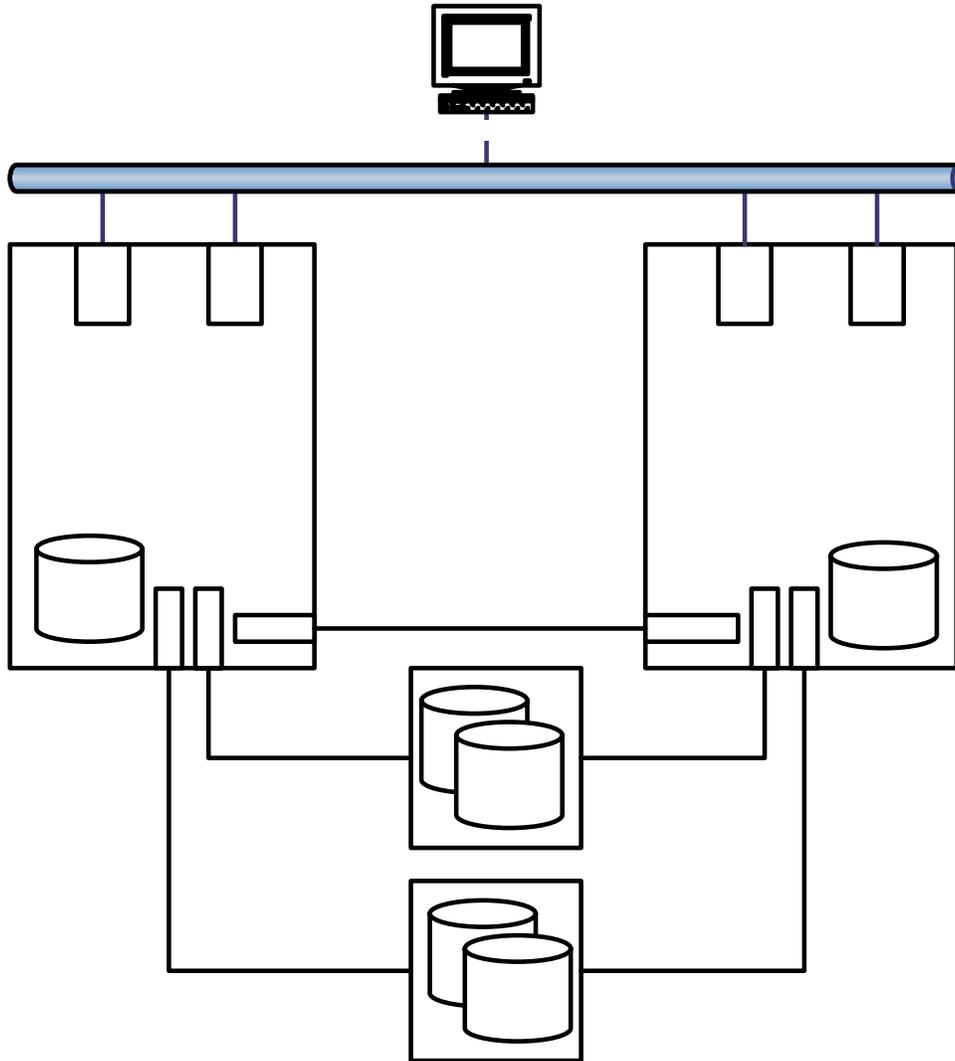
The Topology portion of an HACMP Cluster consists of Nodes, Networks and Network Interfaces (IF in the diagram above). A Node is an instance of an AIX Operating system and the necessary hardware to support it.

A **Node** could be a stand alone System P or other AIX capable hardware. It could also be a logical partition (lpar) in a Power 4 or Power 5 System. If using lpars for nodes it is prudent to assure that nodes expected to back each other up be situated in different managed systems and frames in order to prevent the hardware supporting the nodes from being a single point of failure. The nodes in a cluster do not have to be identical; though they must have the same release of AIX and HACMP installed, they can be of different size with regard to any of the ways a system might be measured so long as the basic

IF

requirements for a cluster node and the requirements for the applications to be supported are available. With HACMP 5.1 or later, you can have up to 32 nodes in a cluster; however, the management of the cluster becomes significantly more challenging as the number of nodes increases, so large clusters should be constructed only when the benefit of the design seems to justify the additional administrative workload.

Networks are of two basic types: IP and non-IP. IP networks can be further categorized as public and private.

> IP networks are used by HACMP for several purposes: client access, heartbeats, cluster manager messages, administrative management of the cluster, and keeping sets of files in synchronization across nodes. Public networks are those which can be used to supply services to clients outside the local network. Private networks are used when intra server application communications are necessary, such as NFS cross mounts within the cluster, or for lock manager traffic when file systems are concurrently accessed by two or more nodes.

> Non-IP networks are used for heartbeat traffic and cluster manager messages. The primary purpose of the non-IP network is to remove the TCP/IP software stack as a single point of failure, to prevent data corruption in a split-brain scenario, should all the IP Networks fail. Three types of non-IP networks are currently available for use in HACMP: tty based serial connection, which is limited by the metrics of the tty hardware; target mode SSA, which is available if SSA subsystems are used for shared volume groups; and heartbeat over disk, which requires Enhanced Concurrent Volume groups, but can be used with any shared disk technology including enterprise storage systems and virtual disks.

Network Interface Cards (NICs) are duplicated in the nodes to prevent a total fallover in case of a failed NIC and to assure heart beating continues. In order to diagnose a NIC failure and determine the failing card it is necessary to set an IP address in a different address subnet on each of the NICs in a node. Each node must have at least two NICs on different subnets for each cluster IP network the node is attached to. This is necessary because the IP subsystems can send a packet out on any NIC that is in the same subnet even though it is not from the NIC with the specified address. In order to assure which NIC a heartbeat packet was sent from it is necessary to keep each heartbeat address on each node in a separate subnet.

Resources

Resources are the things being made highly available. Some examples of resources are: service IP Labels/Addresses, Volume Groups, File Systems, Application Servers, and NFS Exports.

> Client access to servers is provided over IP networks through a **Service Address**, which is a standard TCP/IP network address that can be reached by the client machines, which normally means it is a routable address which will transfer normally through firewalls, routers and gateways. The Service address is normally bound to an application being made highly available, and is not part of the topology. Service

addresses are linked to a service label for resolution purposes. This label is used to identify the association of the address to the Resource Group (discussed in the next section).

The smallest designation of data storage that can be serially shared between nodes (systems) is the **Volume Group**, which by definition is one or more hdisks (hdisks may be either physical or logical disks and will be treated by [LVM](#) the same way without regard to the underlying construction). Volume Groups are associated with a Resource Group using the VG name. Since logical volumes and file systems are sub-constructs of the Volume Group they are normally not specifically identified to HACMP.

An **Application Server** is a logical entity which identifies the full path names of the application start and stop scripts. The name of the application server is the vehicle for adding an application to a Resource Group.

HACMP includes an integrated NFS export mechanism. File Systems to be made highly available NFS exports must be part of a shared volume group, and are associated with a Resource Group using the file system name.

Resource Groups

Resource Groups (RGs) provide three functions: creating a means for managing a set of resources as a single entity, identifying which nodes can activate the resources and determining the policies to be followed when an event that affects the resources occurs.

Resource groups are created by naming them, identifying the nodes they may be activated on in the order of priority, if there is one, and selecting the runtime policies for the RG.

There must be a minimum of two nodes identified with an RG.

The runtime policies fall in three categories: start up policy, fallover policy and fallback policy. Each category has a number of selectable options and sub options.

Dependencies between Resource Groups can also be established. Two layers of parent-child relationship can be specified between RGs without regard to the nodes on which the individual RGs are active. Also, co-location and anti co-location conditions can be specified, along with priorities which resolve conflicts when other directives conflict. The use of these features allows for automated control of very complex application interactions.

Avoiding Single Points of Failure

As stated earlier in this document, HACMP works through the principle of eliminating single points of failure (SPOF). Once an environment has been created where facilities exist to allow use of more than one component to accomplish each task (or at least most tasks), HACMP can be configured to take advantage of the redundancy. It is possible to create a cluster that is still hampered by single points of failure - in fact all clusters eventually resolve to some point of failure that is so expensive that the cost of duplication cannot be justified. However, given clear understanding of the goals and care in planning, a large percentage of the SPOFs in most systems can be eliminated within the cost justification parameters. Let's take another look at the diagram we saw earlier when discussing Cluster Topology.
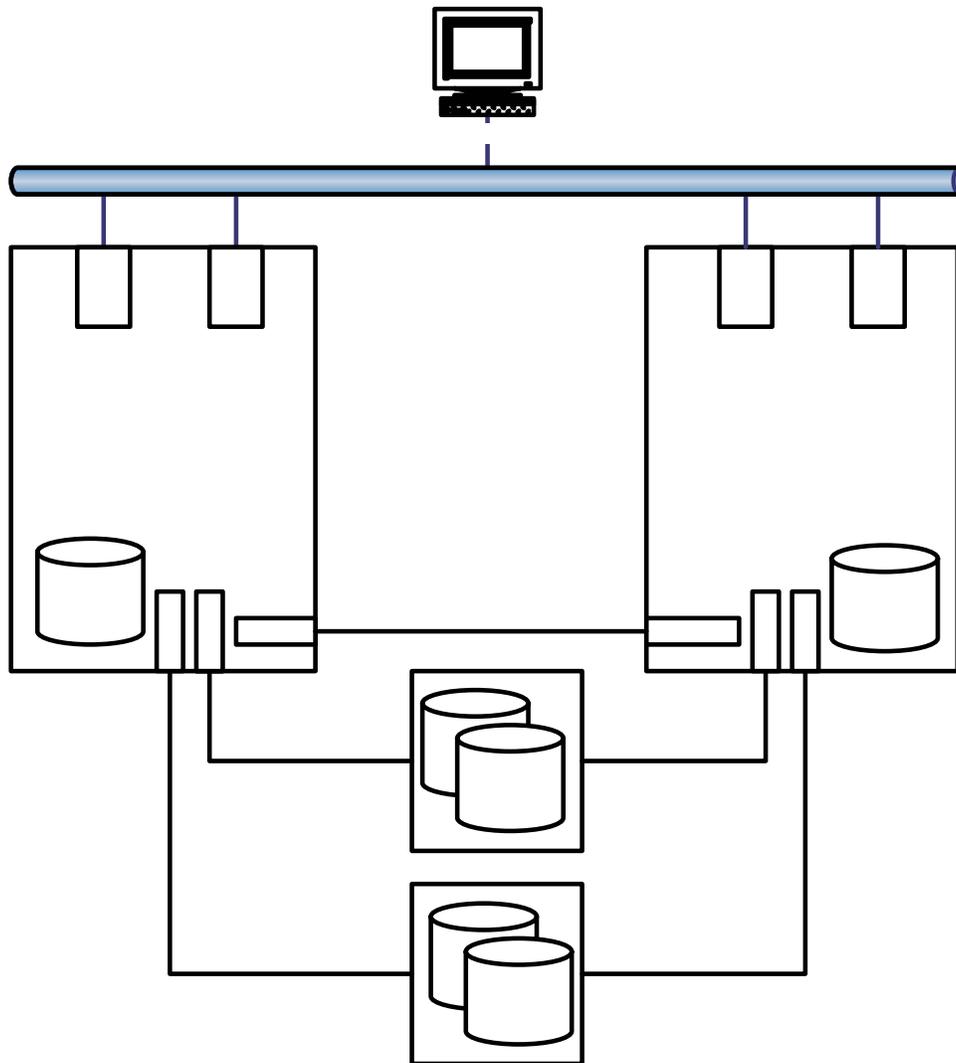
The diagram illustrates several areas in which avoidance of single points of failure is important:

- Two or more nodes are required in order to construct a cluster.

- Every resource group must identify at least two nodes where it can be active.

- Network components must be duplicated in the nodes and entire networks could be duplicated if the justification existed.

- Different protocols for networking (IP and non-IP) are employed to eliminate the failure of heartbeats due to protocol software failure.

- Application data and access to that data should be replicated across all nodes that may activate the application.

- Administrators need reliable access to individual nodes and thereby the cluster, regardless of the state of the cluster as a whole or of individual nodes.

- There is infrastructure in any computer room supporting the systems which, if failed, could cause processing to stop.

TCP/IP Considerations for High Availability

When describing **networks** in the discussion of Topology, we explained the need for multiple NICs and subnets on each node for every IP network, and the importance of one or more non-IP networks, to prevent split brain conditions, in the event of total IP network failure caused by hardware or software. Two subnets with IP addresses for each node must be created for each IP network. In addition it is important to make sure that the network infrastructure itself is as robust as possible. Access to clients should be available through more than one router, and multiple IP networks should not share the same switches or should all have redundant switches.

Either an additional NIC which is not managed by HACMP should be installed in each node or a persistent IP alias should be configured, using HACMP, to provide unblocked access for administrators whether HACMP is active on the node or not. Service addresses and persistent alias addresses may be in the same subnet, but may not be in either of the heartbeat subnets. There is a limit of one persistent alias per node on each IP network.

Additionally, non-IP networks must be created to avoid the IP protocol becoming a single point of failure.

LVM Considerations for High Availability

In order to access **data** from more than one node (usually not at the same time) HACMP relies on the AIX Logical Volume Manager (LVM). LVM organizes disk drives (actually hdisk instances) into volume groups.

> Each disk in the volume group is partitioned into relatively small partitions, all of the same size throughout the VG, and known as physical partitions (pp). This provides a manageable resource which LVM can allocate to logical volumes for use in storing data. Logical volumes are constructed using logical partitions (lp) which are the same size as the physical partitions they will be mapped to. Partition size is a Volume Group characteristic, and is always a power of 2 in megabytes. LVM can create logical volumes by mapping lps to pps without concern for the location of the individual pps in the volume group. Logical volumes can span physical volumes and are not required to be contiguous. LVM has the additional ability to create mirrored logical volumes by mapping lps to more than one pp (2 or 3 copies of the mirror are possible). The application is presented with a contiguous logical volume to store and retrieve data. Journaled file systems (jfs and jfs2) are available to further protect data integrity, or the LVs can be used as raw disk space. File systems and logical volumes can be resized while in use without loss or corruption of data.

> Volume Group configuration information is stored in the AIX ODM object classes. Additionally, each hdisk instance in the volume group carries a Volume Group Descriptor Area which contains complete information about the physical and logical construction of the volume group, including the mapping information of logical partitions to physical partitions, with or without mirroring. Once the volume group has been constructed, it can be connected to another system and the ODM objects created on the new system, by importing information from the Volume Group Descriptor Area on one of the hdisks in the VG.

Since maintaining high availability may require the application to run on one of two or more systems, it is important that each of the candidate systems have connectivity to the disk structures comprising the Shared Volume Group. Furthermore, storage systems must not be physically part of any of the systems accessing them. Shared storage must be in storage systems independent of the system hardware; even the sharing of a frame with a system makes the frame power supply a single point of failure. Additionally, while raid 5 arrays or storage subsystem mirroring provide a means of protecting data access from a disk failure during operation, using either of these methods, instead of LVM mirroring, creates a single point of failure should the storage array fail.

By using the LVM mirroring function and creating the storage in multiple external arrays (minimum of two recommended), shared volume groups can be built across arrays allowing the mirroring of logical volumes with mirror copies in different storage arrays. Using this approach, the loss of an entire storage array will not prevent access to the application data. Notice in the diagram that each array is accessed through a different storage adapter on each node, further reducing single points of failure.

Infrastructure considerations for High Availability

Systems and storage arrays have multiple power supplies; however, connecting them to the same power feed creates a single point of failure at the power connection. In situations where down-time is extremely costly, multiple uninterruptible power sources may be justified. If using multiple ups systems, it is often necessary to purchase an additional feature to allow power fallover to occur in case of a ups failure. Some companies have requested separate power feeds from different power grid locations be installed to avoid the loss of a building power feed being a SPOF. These may be extreme measures for most companies, but running power to the computer room from different circuit breakers, or breaker panels, is a good idea. Identify the connection points to the different circuits, and make sure duplicated devices are not sharing the same power source.

Similar actions should be taken for computer room cooling and other infrastructure facilities.

For more detailed information please refer to the **HACMP for AIX 5L Planning Guide** (SC23-4861), which is available from IBM in printed or PDF format.

HACMP documentation can be found on line at the IBM web site HACMP Library.